

CLAIMS

We claim:

1. A method of identifying an interface of an application program comprising the interface and source code, said method comprising the steps:

parsing the source code of the application program to identify meta information;

storing in a repository the meta information and a link pointing to an original location of the meta information within the application program;

allowing a user to query the repository to determine which source files and which interfaces comprise the application program;

constructing a new source file containing the interfaces which comprise the application program;

storing the new source file and a link pointing to a location of the new source file in the repository; and

constructing a meta language document containing a description of the application program interfaces to enable a connector building tool to build an interface to the application program.

2. The method of claim 1 further comprising the step of:

for a transaction contained in the application program, displaying a visual indicia which navigates via the link to an entry point of the source code corresponding to transaction.

1 3. The method of claim 2 further comprising the step of:
2 displaying the transaction contained in the application program together with a
3 visual indicia which navigates to documentation stored in the repository corresponding
4 to the transaction.

1 4. The method of claim 3 further comprising the steps of:
2 allowing the user to select the transaction;
3 displaying a data structure corresponding to the selected transaction;
allowing the user to select the data structure; and
for the selected data structure, computing input and output fields and overlaying
the fields on the selected data structure.

5. The method of claim 4 further comprising the steps of:
allowing the user to edit the computed fields; and
3 analyzing the selections and editions to determine if an error exists.

6. A method of identifying an interface of an application program comprising the interface and source code, said method comprising the steps:

parsing the source code of the application program to identify meta information;
storing in a repository the meta information and a link pointing to an original location of the meta information within the application program;

allowing a user to query the repository to determine which source files and which interfaces comprise the application program;

for a transaction contained in the application program, displaying a visual indicia which navigates via the link to an entry point of the source code corresponding to transaction;

displaying the transaction contained in the application program together with a visual indicia which navigates to documentation stored in the repository corresponding to the transaction;

allowing the user to select the transaction;

displaying a data structure corresponding to the selected transaction;

allowing the user to select the data structure; and

for the selected data structure, computing input and output fields and overlaying the fields on the selected data structure;

allowing the user to edit the computed fields;

analyzing the selections and editions to determine if an error exists.

constructing a new source file containing the interfaces which comprise the application program;

storing the new source file and a link pointing to a location of the new source file in the repository; and

constructing a meta language document containing a description of the application program interfaces to enable a connector building tool to build an interface to the application program.

1 7. An article of manufacture for use in a computer system for identifying an interface of an
2 application program comprising the interface and source code, said article of manufacture
3 comprising a computer-readable storage medium having a computer program embodied in said
4 medium which causes the computer system to execute a method comprising the steps of:

5 parsing the source code of the application program to identify meta information;

6 storing in a repository the meta information and a link pointing to an original
7 location of the meta information within the application program;

8 allowing a user to query the repository to determine which source files and
9 which interfaces comprise the application program;

10 constructing a new source file containing the interfaces which comprise the
11 application program;

12 storing the new source file and a link pointing to a location of the new source
13 file in the repository; and

14 constructing a meta language document containing a description of the
15 application program interfaces to enable a connector building tool to build an interface
16 to the application program.

8. The article of manufacture of claim 7 wherein the embodied computer program further
causes the computer system to execute the method step:

for a transaction contained in the application program, displaying a visual
indicia which navigates via the link to an entry point of the source code corresponding
to transaction.

1 9. The article of manufacture of claim 8 wherein the embodied computer program further
2 causes the computer system to execute the method step:

3 displaying the transaction contained in the application program together with a
4 visual indicia which navigates to documentation stored in the repository corresponding
5 to the transaction.

1 10. The article of manufacture of claim 9 wherein the embodied computer program further
2 causes the computer system to execute the method steps:

3 allowing the user to select the transaction;

4 displaying a data structure corresponding to the selected transaction;

5 allowing the user to select the data structure; and

6 for the selected data structure, computing input and output fields and overlaying
7 the fields on the selected data structure.

8 11. The article of manufacture of claim 10 wherein the embodied computer program further
9 causes the computer system to execute the method steps:

10 allowing the user to edit the computed fields; and

11 analyzing the selections and editions to determine if an error exists.

12. A article of manufacture of identifying an interface of an application program comprising the interface and source code, said article of manufacture comprising a computer-readable storage medium having a computer program embodied in said medium which causes the computer system to execute a method comprising the steps of:

parsing the source code of the application program to identify meta information;
storing in a repository the meta information and a link pointing to an original location of the meta information within the application program;

allowing a user to query the repository to determine which source files and which interfaces comprise the application program;

for a transaction contained in the application program, displaying a visual indicia which navigates via the link to an entry point of the source code corresponding to transaction;

displaying the transaction contained in the application program together with a visual indicia which navigates to documentation stored in the repository corresponding to the transaction;

allowing the user to select the transaction;

displaying a data structure corresponding to the selected transaction;

allowing the user to select the data structure; and

for the selected data structure, computing input and output fields and overlaying the fields on the selected data structure;

allowing the user to edit the computed fields;

analyzing the selections and editions to determine if an error exists.

constructing a new source file containing the interfaces which comprise the application program;

storing the new source file and a link pointing to a location of the new source file in the repository; and

constructing a meta language document containing a description of the application program interfaces to enable a connector building tool to build an interface to the application program.

1 13. A computer system for identifying an interface of an application program comprising
2 the interface and source code, said computer system comprising:

3
4 a parser for parsing the source code of the application program to identify meta
5 information;

6 storage for storing in a repository the meta information and a link pointing to an
7 original location of the meta information within the application program;

8 a query the repository to determine which source files and which interfaces
9 comprise the application program;

10 a new source file containing the interfaces which comprise the application
11 program;

12 storage for storing the new source file and a link pointing to a location of the
13 new source file in the repository; and

14 a meta language document containing a description of the application program
15 interfaces to enable a a connector building tool to build an interface to the application
16 program.

1 14. The computer system of claim 13 further comprising:

2 a visual indicia, for a transaction contained in the application program, which
3 navigates via the link to an entry point of the source code corresponding to transaction.

1 15. The computer system of claim 14 further comprising:
2 a visual indicia, displayed with the transaction contained in the application
3 program, which navigates to documentation stored in the repository corresponding to
4 the transaction.

1 16. The computer system of claim 15 further comprising:
2 a selected transaction;
3 a display of a data structure corresponding to the selected transaction;
4 a selected data structure;
5 computed input and output fields for the selected data structure; and
6 an overlay of the fields on the selected data structure.

1 17. The computer system of claim 16 further comprising:
2 an editor for editing the computed fields; and
3 an analyzer for analyzing the selections and editions to determine if an error
4 exists.

1 18. A computer system for identifying an interface of an application program comprising
2 the interface and source code, said computer system comprising:
3 a parser for parsing the source code of the application program to identify meta
4 information;
5 storage for storing in a repository the meta information and a link pointing to an
6 original location of the meta information within the application program;
7 a query of the repository to determine which source files and which interfaces
8 comprise the application program;
9 a display of a visual indicia, for a transaction contained in the application
10 program, which navigates via the link to an entry point of the source code
11 corresponding to transaction;
12 a display of a visual indicia, together with the transaction contained in the
13 application program, which navigates to documentation stored in the repository
14 corresponding to the transaction;
15 a selected transaction;
16 a display of a data structure corresponding to the selected transaction;
17 a selected data structure;
18 computed input and output fields for the selected data structure;
19 an overlay of the fields on the selected data structure;
20 an editor for editing the computed fields;
21 an analyzer for analyzing the selections and editions to determine if an error
22 exists.
23 a new source file containing the interfaces which comprise the application
24 program;
25 storage for storing the new source file and a link pointing to a location of the
26 new source file in the repository; and
27 a meta language document containing a description of the application program
28 interfaces to enable a connector building tool to build an interface to the application
29 program.